

LISTING OF THE CLAIMS:

1. (Currently Amended) A method of converting text in a delimited flat file to text in a markup language specified by a document type definition file, said method comprising the steps of:

providing a delimited flat file having text and columns with headings;

providing a map file conforming to said document type definition file and having tags and attributes including references matching said headings;

forming a tree structure from said map file for mapping said text from said flat file into a defined format in said markup language file, and wherein each tag represents one or more nodes of said tree;

traversing ~~node-by-node~~ said nodes of said tree structure, node-by-node, and for each said node entering said attributes into said markup language file; and

when said attributes include one of said references, retrieving text from one of said columns with one of said matching headings of said flat file and entering said text into said markup language file.

2. (Original) A method according to Claim 1, wherein the step of providing a map file includes the step of identifying in the map file references matching all of the column headings of the flat file.

3. (Original) A method according to Claim 1, wherein the providing step includes the step of providing the map file with default text for certain elements and attributes in the markup language file.

4. (Original) A method according to Claim 3, further comprising the step of entering the default text into the markup language for attributes having references that do not match headings of the flat file.

5. (Original) A method according to Claim 1, wherein the flat file is a tab delimited flat file.

6. (Currently Amended) A system for converting text in a delimited flat file to text in a markup language specified by a document type definition file, said system comprising:

means for providing a delimited flat file having text and columns with headings;

means for providing a map file conforming to said document type definition file and having tags and attributes including references matching said headings;

means for forming a tree structure from said map file for mapping said text from said flat file into a defined format in said markup language file, and wherein each tag represents one or more nodes of said tree;

means for traversing ~~node-by-node~~ said nodes of said tree structure, node-by-node, and for each said node entering said attributes into said markup language file; and

means for retrieving text from one of said columns with one of said matching headings of said flat file, when said attributes include one of said references, and for entering said text into said markup language file.

7. (Original) A system according to Claim 6, wherein the map file identifies references matching all of the column headings of the flat file.

8. (Original) A system according to Claim 6, wherein the map file includes default text for certain elements and attributes in the markup language file.

9. (Original) A system according to Claim 8, wherein the traversing means includes means for entering the default text into the markup language for attributes having references that do not match headings of the flat file.

10. (Original) A system according to Claim 6, wherein the flat file is a tab delimited flat file.

11. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for converting text in a delimited flat file to text in a markup language specified by a document type definition file, said method steps comprising:

providing a delimited flat file having text and columns with headings;

providing a map file conforming to said document type definition file and having tags and attributes including references matching said headings;

forming a tree structure from said map file for mapping said text from said flat file into a defined format in said markup language file, and wherein each tag represents one or more nodes of said tree;

traversing ~~node-by-node~~ said nodes of said tree structure, ~~node-by-node~~, and for each said node entering said attributes into said markup language file; and

when said attributes include one of said references, retrieving text from one of said columns with one of said matching headings of said flat file and entering said text into said markup language file.

12. (Original) A program storage device according to Claim 11, wherein the step of providing a map file includes the step of identifying in the map file references matching all of the column headings of the flat file.

13. (Original) A program storage device according to Claim 11, wherein the providing step includes the step of providing the map file with default text for certain elements and attributes in the markup language file.

14. (Original) A program storage device according to Claim 13, further comprising the step of entering the default text into the markup language for attributes having references that do not match headings of the flat file.

15. (Original) A program storage device according to Claim 11, wherein the flat file is a tab delimited flat file.

16. (New) A method according to Claim 1, wherein all of said headings of said columns of said flat file have matching references in said map file to help enter all of said text of said flat file into said markup language file.